Supervisory Control of Real-Time Systems  
Using Prioritized Synchronization

Ratnesh Kumar\textsuperscript{1} and Mark A. Shayman\textsuperscript{2}

\textsuperscript{1} Department of Electrical Engineering  
University of Kentucky  
Lexington, KY 40506-0046  
Email: kumar@engr.uky.edu

\textsuperscript{2} Department of Electrical Engineering and ISR  
University of Maryland  
College Park, MD 20742  
Email: shayman@src.umd.edu

\textbf{Abstract.} The theory of supervisory control of discrete event systems is extended to the real-time setting. The real-time behavior of a system is represented by the set of all possible timed traces of the system. This is alternatively specified using timed automata where each transition is associated with an event occurrence time set during which time the transition can occur. Our model for time is more general in that the time advances continuously as compared to a model where time advances discretely. We extend the notion of prioritized synchronous composition to the real-time setting to use it as the control mechanism. It is shown that a suitable extension of the controllability condition to the real-time setting yields a condition for the existence of a supervisor achieving a desired timed behavior. Although the real-time controllability is similar in form to its untimed counterpart, they are different in the sense that one does not imply the other and vice-versa.

\textbf{Keywords:} Discrete event systems, timed automata, real-time systems, prioritized synchronization, supervisory control, real-time controllability, real-time relative-closure

1 \textbf{Introduction}

Control of logical or untimed behavior of discrete event systems has extensively been studied since its initiation by Ramadge-Wonham [20]. Refer for example to the survey articles [21, 24] and the book [14]. The untimed behavior of a system retains the event ordering information but ignores the event timing

\textsuperscript{*} This research was supported in part by the Center for Robotics and Manufacturing, University of Kentucky, in part by the National Science Foundation under the Grants NSF-CDR-8803012, NSF-ECS-9409712, NSF-ECS-9312587, the Minta Martin Fund for Aeronautical Research, and the General Research Board at the University of Maryland.
information. The event timing information must also be retained when the design objective involves real-time constraints such as in a communication system where a transmitted message should be received within a fixed time delay. This paper deals with an extension of the supervisory control theory to the real-time setting.

Sets of timed traces, i.e., finite sequences of pairs, each consisting of an event and its occurrence time, is used to describe the real-time behavior of a discrete event system. This is alternatively represented using timed automata where each transition is associated with an event occurrence time set. Each event has a timer (a stop-watch) which advances when the automaton is in an activity state where the event is enabled; the transition occurs when the event timer lies within the occurrence time set of the transition. Timed automaton model of the real-time behavior of a system is appealing since from the work of Alur-Dill [3, 2], its associated “region automaton” is finite so that the emptiness of the set of its accepted timed traces can be algorithmically verified.

There has been extensive prior work on verification of real-time systems. This includes the work of Ostroff-Wonham on real-time temporal logic [17], Alur-Dill on timed automata [3], Reed-Roeoe [22] on timed CSP, Coolahan-Roussopoulos [8] and Berthomieu-Diaz [4] on timed Petri-nets, Jahanian-Mok on safety analysis [12], etc.

The control and synthesis of supervisors for the real-time systems has received some attention recently. In the work of Brave-Heymann [7] and also in the work of Brandin-Wonham [5] a discretized model for time has been used: the time advances when a “tick” transition of the timer occurs. Such a model is not as general as the ones in which a dense model for time is used [1, Section 2.3.2]. Furthermore, the possibility of pre-empting a “tick” transition by a forcible event as in the work of Brandin-Wonham [5] is an artifact arising due to the discretized model of time, and is unrealistic. The work of Wong-Toi-Hoffman [25] uses a dense model of time, and we extend their work to consider control by way of prioritized synchronous composition (PSC) of plant and supervisor.

In this paper we represent real-time behavior using timed automaton which uses a dense model of time. The control is achieved by prioritized synchronous composition of a plant and a supervisor timed automaton. We have extended the notion of prioritized synchronization first given by Heymann [9] and later used by Shayman-Kumar [23] for the untimed systems to the real-time setting. We show that the prioritized synchronous composition is associative and under certain mild conditions which hold in the supervisory control setting can be reduced to the strict synchronous composition by using the technique of augmentation.

The conditions for the existence of a supervisor achieving a given closed-loop specification are the natural generalizations of the controllability and relative-closure conditions for the untimed case. So the earlier results such as the existence of a unique maximally permissive supervisor, etc., automatically follow. The algorithmic tests and computations however differ from the untimed case but are conceptually similar. In order to illustrate the differences and similarities we give an algorithmic test for the real-time controllability. This exploits the
finiteness of the “region automaton” associated with a timed automaton. The real-time controllability condition is similar in form to its untimed counterpart, however, they are unrelated in the sense that neither of them imply the other.

The rest of the paper is organized as follows: In Section 2 we introduce our notation. Section 3 defines the timed automaton and the associated timed trace model. In Section 4 we define the prioritized synchronous composition of timed automaton and also of timed traces and establish their consistency and associativity, whereas in Section 5 we obtain its reduction to strict synchronous composition using the notion of augmentation. Section 6 deals with the supervisory control issues and introduces the notion of controllability and relative-closure for the real-time setting. Finally we conclude our work in Section 7.

2 Notation and Preliminaries

We use $\Sigma$ to denote the set of events, and $\Sigma^*$ to denote the set of all finite length event sequences, called traces. $e$ is used to denote the zero length trace. For a trace $s$, $|s|$ denotes its length; $pr(s) \subseteq \Sigma^*$ denotes the set of all prefixes of $s$; and for $k \leq |s|$, $s^k$ denotes the prefix of length $k$ of $s$. If a trace $s_1$ is a prefix of a trace $s_2$, then it is denoted as $s_1 \leq s_2$; if $s_1$ is a proper prefix of $s_2$, then it is written as $s_1 < s_2$.

A timed trace is an element of $(\Sigma \times \mathbb{R}_+)^*$. Thus given a timed trace $e$, it is of the form:

$$e = (\sigma_1(e), t_1(e))(\sigma_2(e), t_2(e)) \ldots (\sigma_{|e|}(e), t_{|e|}(e)),$$

where $\sigma_i(e) \in \Sigma$ and $t_i(e) \in \mathbb{R}_+$ are the $i$th event and its corresponding occurrence time, and $|e|$ denotes the length of $e$. By definition the sequence of event occurrence times $\{t_i(e)\}$ is monotonically increasing. The notation $tr(e) := \sigma_1(e)\sigma_2(e) \ldots \sigma_{|e|}(e)$ is used to denote the associated “untimed” trace of $e$. We use $pr(e)$ to denote the set of all prefixes of $e$; and for $k \leq |e|$, $e^k$ denotes the prefix of length $k$ of $e$. The notation $(e,0)$ is used to denote the timed trace of length zero. Given two timed traces $e$ and $f$, we use $e \leq f$ (respectively, $e < f$) to denote that $e$ is a prefix (respectively, a proper prefix) of $f$.

3 Timed Automata and Timed Trace Model

We use a pair of timed trace sets to describe the real-time behavior of a discrete event system. The set of generated timed traces consists of all sequences of pairs consisting of an event and its occurrence time that are possible in the system; a subset of it is the set of accepted timed traces whose execution results in completion of certain tasks.

A timed automata is a six tuple:

$$P := (X_P, \Sigma_P, \delta_P, F_P, x^0_P, X^m_P),$$
where $X_P$ is the finite set of activity states, $\Sigma$ is the finite set of events, $\delta_P : X_P \times \Sigma \rightarrow 2^{X_P}$ is the activity state transition function, $T_P : X_P \times \Sigma \times X_P \rightarrow \mathbb{R}^+$ is the occurrence time set function, $x_0^P \in X_P$ is the initial activity state, and $X_P^F \subseteq X_P$ is the set of final activity states. A triple $(x, \sigma, x') \in X_P \times \Sigma \times X_P$ is called an activity state transition if $x' \in \delta_P(x, \sigma)$. For each activity state transition $(x, \sigma, x')$, we require that its occurrence time set $T_P(x, \sigma, x')$ be a finite union of intervals with rational end-points. If $(x, \sigma, x')$ is not an activity state transition, then we assume that the associated occurrence time set is empty, i.e., $T_P(x, \sigma, x') = \emptyset$. For $x \in X_P$, we define $\Sigma_P(x) := \{ \sigma \in \Sigma : \delta_P(x, \sigma) \neq \emptyset \}$ to be the set of events defined at state $x$ in $P$.

In order to describe the evolution of the system as the time advances, a timer is associated with each event. We use $\tau \in \mathbb{R}_+^{[\Sigma]}$ to denote the timer vector and $\tau(\sigma) \in \mathbb{R}_+$ to denote the timer associated with event $\sigma$. When the system is in an activity state where a transition on a certain event is defined, then the timer for the associated event advances as the time elapses, and thus keeps track of the enablement time of the event. If the event enablement time lies within the occurrence time set of the associated transition, then the transition can occur resulting in a change of the activity state. The event timer is reset to zero whenever the transition occurs.

The state of the system is a pair $(x, \tau) \in X_P \times \mathbb{R}_+^{[\Sigma]}$ consisting of an activity state $x$ and a timer vector $\tau$. An event $\sigma$ can occur at state $(x, \tau)$ if there is an activity state transition on event $\sigma$ defined at state $x$ and the event timer value $\tau(\sigma)$ lies within the occurrence time set associated with that transition. Given $\Sigma' \subseteq \Sigma$, we use $\tau \downarrow_{\Sigma'}$ to denote the timer vector obtained by resetting each of the timers associated with events in $\Sigma'$ to zero; and given $t \in \mathbb{R}_+$, $\tau \oplus t$ is used to denote the timer vector obtained by adding $t$ to each of the timers associated with events in $\Sigma$.

In this paper we only study systems that are deterministic, i.e., systems in which the next state is uniquely determined by the current state and the event that occurs in that state. It is not difficult to verify that for a timed automaton to represent a deterministic system the following should hold: Whenever there is a nondeterministic activity state transition on a certain event in the timed automaton, the occurrence time sets associated with those transitions are disjoint. Hence we have the following definition of a deterministic timed automaton:

**Definition 1** A timed automaton $P$ is said to be deterministic if

$$\forall x \in X_P, \sigma \in \Sigma : \exists x_1, x_2 \in \delta_P(x, \sigma), x_1 \neq x_2 = T_P(x, \sigma, x_1) \cap T_P(x, \sigma, x_2) = \emptyset.$$

The activity state transition function of $P$ can be extended to obtain the deterministic state transition function $\overline{\delta}_P : X_P \times \Sigma \rightarrow X_P$ as follows ($(x, \tau) \in X_P$):

$$\overline{\delta}_P ((x, \tau), \sigma) := \begin{cases} (x', \tau \downarrow_{(\Sigma \setminus \Sigma_P(x)) \cup \{\sigma\}}) & \exists x' \in \delta_P(x, \sigma) : \tau(\sigma) \in T_P(x, \sigma, x') \\ \text{undefined} & \text{otherwise} \end{cases}$$

Thus when activity state is $x$ and the event timer $\tau(\sigma)$ lies within the interval $T_P(x, \sigma, x')$ for some $x' \in \delta_P(x, \sigma)$ (note that by the requirement of determinism
there exists at most one such \( x' \)), then the event \( \sigma \) can occur. This results in the new activity state \( x' \), and also in the resetting of the event timer for \( \sigma \) as well as of the event timers for all those events that are no more defined at the activity state \( x' \).

The state transition function can be extended from events to timed traces inductively as follows (the extension is denoted by \( \overrightarrow{\delta} \)):

\[
\overrightarrow{\delta}_P ((x, \tau), (e, 0)) := (x, \tau) \\
\overrightarrow{\delta}_P ((x, \tau), (\sigma, t)) := \overrightarrow{\delta}_P (\delta_P (\overrightarrow{\delta}_P ((x, \tau) \oplus \Sigma_P (\sigma) t), \sigma)) \\
\overrightarrow{\delta}_P ((x, \tau), e(\sigma, t)) := \overrightarrow{\delta}_P (\overrightarrow{\delta}_P ((x, \tau), e(\sigma, t)))
\]

This extended transition function can be used to define the set of timed traces \( \text{generated} \) and \( \text{accepted} \) by \( P \), denoted \( T(P) \) and \( \text{T}^m(P) \), respectively:

\[
T(P) := \{ e \mid \overrightarrow{\delta}_P ((x^0_P, 0), e) \text{ defined} \} ; T^m(P) := \{ e \in T(P) \mid \overrightarrow{\delta}_P ((x^0_P, 0), e) \in X^m \}
\]

The pair \( (T^m(P), T(P)) \) is called the timed trace model of \( P \). It is clear that

\[
p(T^m(P)) \subseteq T(P) = p(T(P)) \neq \emptyset. \tag{1}
\]

The system is said to be non-blocking if instead of the containment, the equality holds in (1).

**Remark 1** It is clear from the definition of the timed trace model of \( P \) that a transition of \( P \) whose occurrence time set is a union of say \( i \) disjoint intervals can be replaced by \( i \) transitions, each with occurrence time set a unique interval, without altering the time trace model of \( P \). So the definition of the timed automaton given above is a special case of that given by Alur-Dill [3, 2], where more general timers (that are not necessarily associated with events) and more general timer constraints (that are not necessarily constraints on unique timers) are permitted. Hence using a result of Alur-Dill [3, 2], it follows that the “region automaton” associated with \( P \) is finite, and hence the emptiness of its generated or accepted timed traces can be algorithmically determined.

## 4 Prioritized Synchronous Composition

Prioritized synchronous composition (PSC) has been proposed by Heymann [9] and Heymann-Meyer [10] as a suitable mechanism of control, and it has been applied for the control of nondeterministic discrete event systems modeled using trajectory sets or refusal trace sets in [23, 15, 16]. Here we extend the definition of prioritized synchronization to the real-time setting.

For prioritized synchronous composition each system is assigned a priority set of events, and for an event to occur in the composed system each sub-system having priority over the event must participate. If an event belongs only to the priority set of one system, then the event occurs asynchronously (without the participation of the other system) if the other system cannot participate in its occurrence, otherwise it occurs synchronously. This kind of synchronization is
called broadcast synchronization. Note that in the special case when the two priority sets are identical to the entire event set, prioritized synchronous composition reduces to strict synchronous composition (SSC) [11] where all events always occur synchronously.

In the context of supervisory control, the priority set of the plant includes the controllable and the uncontrollable events, whereas the priority set of the supervisor includes the controllable and the driven events. Since controllable events are common to the two priority sets, they occur synchronously in plant and supervisor, which is consistent with the fact that their occurrence in the closed-loop system is controlled by the supervisor. Uncontrollable events can occur asynchronously in the plant which is consistent with the fact that their occurrence cannot be prevented by the supervisor; however, the supervisor can track their occurrence by means of broadcast synchronization. Driven events are the duals of the uncontrollable events, and their occurrence requires the participation of the supervisor but not necessarily synchronization by the plant.

Next we formalize the notion of prioritized synchronization in the real-time setting by appropriately extending the definition given by Heymann.

**Definition 2** Letting $A \subseteq \Sigma$ and $B \subseteq \Sigma$ denote the priority sets of timed automata $P$ and $Q$ respectively, their prioritized synchronous composition (PSC), denoted $P_A ||_B Q$, is the timed automaton $R$, where $X_R := X_P \times X_Q; x^0_R := (x^0_P, x^0_Q)$, and $X^m_R := (X^m_P \times X^m_Q)$. The activity transition function is defined as follows $(x_r, (x_p, x_q), x'_r, (x'_p, x'_q)) \in X_R, \sigma \in \Sigma$:

$$
\delta_R (x_r, \sigma) := \begin{cases} 
\delta_P (x_p, \sigma) \times \delta_Q (x_q, \sigma) & \text{if } \delta_P (x_p, \sigma), \delta_Q (x_q, \sigma) \neq \emptyset \\
\delta_P (x_p, \sigma) \times \{ x_q \} & \text{if } \delta_P (x_p, \sigma) \neq \emptyset, \sigma \notin B \\
\emptyset & \text{otherwise}
\end{cases}
$$

The occurrence time set function is defined as follows $(x_r, (x_p, x_q), x'_r, (x'_p, x'_q)) \in X_R$ and $\sigma \in \Sigma$:

$$
T_R (x_r, \sigma, x'_r, (x'_p, x'_q)) := \begin{cases} 
T_P (x_p, \sigma, x'_p) \cap T_Q (x_q, \sigma, x'_q) & \text{if } x'_r \in \delta_P (x_p, \sigma) \times \delta_Q (x_q, \sigma) \\
T_P (x_p, \sigma, x'_p) - \bigcup_{x_q' \in \delta_Q (x_q, \sigma)} T_Q (x_q, \sigma, x'_q) & \text{if } x'_r \in \delta_P (x_p, \sigma) \times \{ x_q \} \\
T_Q (x_q, \sigma, x'_q) & \text{if } x'_r \in \{ x_p \} \times \delta_Q (x_q, \sigma) \\
T_Q (x_q, \sigma, x'_q) - \bigcup_{x_p' \in \delta_P (x_p, \sigma)} T_P (x_p, \sigma, x'_p) & \text{if } x'_r \in \{ x_p \} \times \delta_Q (x_q, \sigma)
\end{cases}
$$

Thus we have three different cases to consider: (i) A transition $(x_r, \sigma, x'_r)$ in $R$ is present whenever there is a transition $(x_p, \sigma, x'_p)$ in $P$ and a transition $(x_q, \sigma, x'_q)$ in $Q$. Since such a transition occurs synchronously, its occurrence time set in $R$ is the intersection of the occurrence time sets of the corresponding transitions in $P$ and $Q$. (ii) If $\sigma$ is not in the priority set of $Q$ but there is a transition $(x_p, \sigma, x'_p)$ in $P$, then a transition $(x_r, \sigma, (x'_p, x_q))$ is present in $R$. Thus if a transition $(x_q, \sigma, x'_q)$ is present in $Q$, then this transition in $R$ of case (ii) is in
addition to the transition at the same state and on same event in \( \mathcal{R} \) of case (i). Since such a transition occurs asynchronously in \( \mathcal{P} \) (when \( \mathcal{Q} \) is unable to participate), the occurrence time set of this transition in \( \mathcal{R} \) is the difference between the occurrence time set of the corresponding transition in \( \mathcal{P} \) and the union over the occurrence time sets of all the transitions labeled \( \sigma \) at state \( x_{q} \) in \( \mathcal{Q} \). (iii) The third case is the dual of the second case and can be understood analogously.

It can be verified from the above definition that the determinism of timed automata is preserved under prioritized synchronous composition, i.e., determinism of \( \mathcal{P} \) and \( \mathcal{Q} \) implies determinism of \( \mathcal{P} \| B \mathcal{Q} \) for any \( A \) and \( B \). The following result establishes another desirable property of prioritized synchronous composition that it is associative.

**Theorem 1** Given timed automata \( \mathcal{P}, \mathcal{Q}, \mathcal{R} \) with priority sets \( A, B, C \) respectively,

\[
\mathcal{P} \| B \mathcal{Q} \mathcal{C} = (\mathcal{P} \| B \mathcal{Q}) \mathcal{A} \mathcal{B} \mathcal{C} \mathcal{R}.
\]

Next in order to obtain a relationship between the timed trace model of the composed system and those of the component systems, we define the prioritized synchronous composition of timed traces.

**Definition 3** Given priority sets \( A \) and \( B \), the prioritized synchronous composition of timed traces \( e_{p} \) and \( e_{q} \), denoted \( e_{p} \| B e_{q} \), is defined inductively on \( |e_{p}| + |e_{q}| \) as follows:

\[
(\sigma, t)_{A} \| B (e, 0) = (e, 0) \| B (\sigma, t) = (\sigma, t)
\]

For \( |e_{p}| + |e_{q}| \geq 1 \), let \( \sigma_{p} := \sigma_{|e_{p}|}, \sigma_{q} := \sigma_{|e_{q}|}, t_{p} := t_{|e_{p}|}, t_{q} := t_{|e_{q}|} \), then \( e_{p} \| B e_{q} := T_{1} \cup T_{2} \cup T_{3} \), where

\[
T_{1} := \begin{cases} 
\{ e(\sigma_{p}, t_{p}) \mid e \in e_{p}^{\text{|}e_{p}| - 1} \| B e_{q} \} & \text{if } |e_{p}| \geq 1, \sigma_{p} \not\in B \\
\emptyset & \text{otherwise}
\end{cases}
\]

\[
T_{2} := \begin{cases} 
\{ e(\sigma_{q}, t_{q}) \mid e \in e_{p} \| B e_{q}^{\text{|}e_{q}| - 1} \} & \text{if } |e_{q}| \geq 1, \sigma_{q} \not\in A \\
\emptyset & \text{otherwise}
\end{cases}
\]

\[
T_{3} := \begin{cases} 
\{ e(\sigma, t) \mid e \in e_{p}^{\text{|}e_{p}| - 1} \| B e_{q}^{\text{|}e_{q}| - 1} \} & \text{if } |e_{p}|, |e_{q}| \geq 1; \sigma_{p} = \sigma_{q} := \sigma; t_{p}, t_{q} := t \\
\emptyset & \text{otherwise}
\end{cases}
\]

The following result states that the prioritized composition of timed event traces defined above is consistent with the definition of the composition of the timed automata.

**Theorem 2** Given timed automata \( \mathcal{P} \) and \( \mathcal{Q} \) with priority sets \( A \) and \( B \) respectively, let \( \mathcal{R} := \mathcal{P} \| B \mathcal{Q} \). Then

\[
T^{m}(\mathcal{R}) = \bigcup_{e_{p} \in T^{m}(\mathcal{P}), e_{q} \in T^{m}(\mathcal{Q})} e_{p} \| B e_{q} ; \quad T(\mathcal{R}) = \bigcup_{e_{p} \in T(\mathcal{P}), e_{q} \in T(\mathcal{Q})} e_{p} \| B e_{q}.
\]

**Remark 2** The result of Theorem 2 can be used to extend the definition of composition of timed traces to timed trace models: Given timed trace models
\((P^n, P)\) and \((Q^n, Q)\) with priority sets \(A\) and \(B\) respectively, their composition is the timed trace model \((R^n, R)\), where
\[
R^n := \bigcup_{e_p \in P^n, e_q \in Q^n} e_p \, A \parallel B \, e_q; \quad R := \bigcup_{e_p \in P, e_q \in Q} e_p \, A \parallel B \, e_q.
\]

It also follows from Theorem 1 that this composition is also associative.

5 Reduction of PSC to SSC by Augmentation

In this section we show that in the special case when the priority sets jointly exhaust the entire event set, prioritized synchronization can be expressed as strict synchronization of each system augmented with those events that belong to the priority set of the other system only.

Given a timed automaton \(\mathcal{P}\) and an augmentation event set \(D \subseteq \Sigma\), the augmentation of \(\mathcal{P}\) with events in \(D\), denoted \(\mathcal{P}^D\), adds a self-loop transition at each activity state on every event in \(D\) and assigns an occurrence time set to it so that the union of the occurrence time sets associated with all \(D\)-labeled transitions at any activity state exhausts the set of positive reals. Thus the augmented system is always able to execute each event in \(D\). So if we augment a given system with events in the sole priority set of another system, then in the composition such events will always occur synchronously since the second system must participate in their occurrence whereas the augmented system can always participate in their occurrence.

Using the definition of augmentation and the associativity of prioritized synchronous composition it can be shown that the PSC can be reduced to SSC of suitably augmented systems when the priority sets jointly exhaust the entire event set.

**Theorem 3** Given timed automata \(\mathcal{P}\) and \(\mathcal{Q}\) with priority sets \(A\) and \(B\), respectively, satisfying \(A \cup B = \Sigma\), we have
\[
\mathcal{P} \, A \parallel_B \, \mathcal{Q} = \mathcal{P}^{\Sigma - A} \parallel_B \, \mathcal{Q}^{\Sigma - B}.
\]

This result is particularly useful in the supervisory control setting since the requirement \(A \cup B = \Sigma\) is naturally satisfied, as each event must belong to the priority set of either plant or supervisor.

**Remark 3** It follows from Theorem 3 that when \(A \cup B = \Sigma\), the timed trace model of the composed system can be obtained by intersecting the timed trace models of the augmented systems. Formally,
\[
T^m(\mathcal{P} \, A \parallel_B \, \mathcal{Q}) = T^m(\mathcal{P}^{\Sigma - A}) \cap T^m(\mathcal{Q}^{\Sigma - B}); \quad T(\mathcal{P} \, A \parallel_B \, \mathcal{Q}) = T(\mathcal{P}^{\Sigma - A}) \cap T(\mathcal{Q}^{\Sigma - B}).
\]
6 Synthesis and Computation of Supervisor

For supervisory control of a discrete event plant, both plant and supervisor are modeled by timed automata. The priority set $A$ of a plant $P$ consists of the controllable and the uncontrollable events, whereas the priority set $B$ of a supervisor $S$ consists of the controllable and the driven events. Thus in this case $A \cup B = \Sigma$, and PSC can be reduced to SSC by using the techniques of augmentation. The control objective is to design a supervisor $S$ such that the controlled plant or the closed-loop system $P_{A \| A} \parallel B S$ meets certain desired constraints specified as a target behavior $K \subseteq (\Sigma \times \mathcal{R}_+)^*$, i.e., it is required that $T^m(P_{A \| A} \parallel B S) = K$. Since the following holds for the target behavior: $K = T^m(P_{A \| A} \parallel B S) = T^m(P_{\Sigma - A} \parallel B S) \cap T^m(Q_{\Sigma - B}) \subseteq T^m(P_{\Sigma - A})$, there is no loss of generality in assuming that $K \subseteq T^m(P_{\Sigma - A})$. Furthermore, it is also desired that any generated timed trace of the controlled plant be a prefix of some timed trace that signifies completion of a task, i.e., a non-blocking supervision is desired which can be specified as:

$$\Pr(T^m(P_{A \| A} \parallel B S)) = T(P_{A \| A} \parallel B S).$$

Next we extend the notions of controllability and relative-closure to the real-time setting and show that these two conditions are jointly necessary and sufficient for the existence of a supervisor achieving the desired behavior.

**Definition 4** Given a plant $P$ with priority set $A$, a priority set $B$ of the supervisor to be designed such that $A \cup B = \Sigma$, and a desired behavior $K \subseteq T^m(P_{\Sigma - A})$:

$K$ is said to be **real-time controllable** if

$$\Pr(K \cap ((A - B) \times \mathcal{R}_+) \cap T(P)) \subseteq \Pr(K).$$

$K$ is said to be **real-time relative-closed** if

$$\Pr(K) \cap T^m(P) \subseteq K.$$

Conditions similar to these were first given by Brave-Heymann [7]. The difference arises due to the discretization as opposed to the dense model of time used in that reference. The following result states that controllability and relative-closure are necessary and sufficient for the existence of a supervisor.

**Theorem 4** Given a plant $P$ with priority set $A$, a priority set $B$ of the supervisor such that $A \cup B = \Sigma$, and a desired behavior $K \subseteq T^m(P_{\Sigma - A})$, there exists a non-blocking supervisor $S$ such that $T^m(P_{A \| A} \parallel B S) = K$ if and only if $(\Pr(K), \Pr(K))$ is a timed trace model, $K$ is real-time controllable and real-time relative-closed. In this case $S$ can be chosen to be any timed automaton with timed trace model $(\Pr(K), \Pr(K))$. 


Remark 4 The result of Theorem 4 is similar to its untimed counterpart. However, it is easy to show that the real-time controllability (respectively, real-time relative-closure) does not imply the untimed controllability (respectively, untimed relative-closure) and vice-versa. A condition is given in [7] under which the real-time controllability is equivalent to its untimed counterpart.

On the other hand, the similarity between the conditions of Theorem 4 and their untimed counterpart can be used to deduce useful conclusions. For example one can conclude that the real-time controllability as well as the real-time relative-closure are preserved under union. So if $K$ does not satisfy the conditions of Theorem 4, then there exists a unique maximal subset of $K$ satisfying the conditions. So a unique \textit{maximally permissive supervisor} exists. Wong-Toi-Hoffman [25] present a technique for its computation by reducing its computation to the untimed setting [19, 6, 13].

Next to illustrate how the existence of a supervisor can be verified, we provide an algorithmic test for the real-time controllability. Using a result from the untimed case [6], it is easy to show that $K$ is real-time controllable if and only if

$$
pr(K)\left[ (A - B) \times \mathbb{R}_+ \right]^* \cap T(\mathcal{P}) \subseteq pr(K),
$$

which is equivalent to

$$
pr(K)\left[ (A - B) \times \mathbb{R}_+ \right]^* \cap T(\mathcal{P}) \cap (pr(K))^c = \emptyset,
$$

where $(pr(K))^c$ denotes the complement of $pr(K)$.

Let $Q$ be a timed automaton with timed trace model $(pr(K), pr(K))$. We first construct a timed automaton which accepts $(pr(K))^c$. Since $Q$ is deterministic such a timed automaton is easily obtained by adding a “dump” activity state, and defining a transition from each activity state on each event to the dump activity state with its occurrence time set being the complement of the union of the time sets for the existing transitions on that event. The dump activity state is the only final activity state. Formally, the complement of $Q$, denoted $Q'$, is the timed automaton:

$$
Q' := (X_Q \cup \{x^d_Q\}, \Sigma, \delta_Q', T_Q', x_Q^0, \{x^d_Q\}),
$$

where $x^d_Q \not\in X_Q$ is the added dump activity state; for each $x \in X_Q \cup \{x^d_Q\}$ and $\sigma \in \Sigma$:

$$
\delta_Q'(x, \sigma) := \delta_Q(x, \sigma) \cup \{x^d_Q\};
$$

and for each transition $(x, \sigma, x''_Q)$ in $Q'$ (but not in $Q$):

$$
T_Q'(x, \sigma, x^d_Q) := \mathbb{R}_+ - \bigcup_{x' \in \delta_Q(x, \sigma)} T_Q(x, \sigma, x'),
$$

whereas the occurrence time sets of the existing transitions remain the same.

Next we construct a timed automaton which accepts $pr(K)\left[ (A - B) \times \mathbb{R}_+ \right]^*$. Again this is obtained by adding a dump activity state in $Q$, and defining a transition from each activity state on each uncontrollable event to the dump.
activity state with its occurrence time set being the complement of the existing transitions on that event. In this case we make all the activity states (including the dummy one) of the resulting timed automaton, say $Q^u$, to be final. Then from (2) it follows that $K$ is real-time controllable if and only if

$$T^m((Q^u \star || P) \star || Q^c) = \emptyset,$$

which can be verified by constructing the region automaton associated with $(Q^u \star || P) \star || Q^c$ and checking the emptiness of its accepted (untimed) language.

7 Conclusion

We have shown that the timed automata model and the prioritized synchronous composition of timed automata provides a suitable framework for supervisory control of real-time discrete event systems. The verification of existence of a supervisor and also its computation remains algorithmic, so an automated synthesis of supervisor is possible. The associativity property of the prioritized synchronous composition suggests that the framework is suitable for modular and decentralized control as well. Results from the untimed setting for control under partial observation using prioritized synchronization [16] can also be easily generalized to the present setting. Finally, although our results are applicable to the systems modeled using timed automaton, they can be generalized to the systems modeled using decidable hybrid automaton [18].

References


This article was processed using the \texttt{\LaTeX} macro package with LLNCS style